

## REMARKS/ARGUMENTS

Claims 1-4 are pending in the present application. By this Amendment, claim 3 is canceled, and claim 1 is amended. Support for the amendments may be found at least on page 9, lines 1-19 and page 12, lines 11-15.

### **I. 35 U.S.C. § 103, Obviousness, Claims 1-4**

The examiner has rejected claims 1-4 under 35 U.S.C. § 103 as being unpatentable over Venkatapathy C et al. ["An introduction to Web Services Gateway", May 2002] (hereinafter "Venkatapathy") in view of Chan ["Web services sub-team report", June 2002] (hereinafter "Chan"). This rejection is respectfully traversed.

The examiner states:

Regarding claim 1, Venkatapathy teaches a method for a web services gateway to enable a web client to access a web service (i.e. web services can be accessed from applications and processes both within the corporate firewall and enables a external user to access to the service,

page 1 lines 6-10, page 2 lines 9-11, and lines 30-36), the method comprising the steps of:

receiving a profile from the web service, the profile containing details of how to communicate with the web service (e.g. provided a StockQuote service that is deployed inside the firewall of your enterprise and sharing the services with the clients, page 2 lines 38-43 and page 3 lines 1-3);

creating a document based on the profile (e.g. importing the WSDL document into the gateway), the document being in a format recognizable to the web client and containing details of how to communicate with the web service via the gateway (i.e. the gateway will generate a new WSDL file that can be shared with the clients, page 3 lines 1-6); and

providing, to a third party (e.g. UDDI directory), information relating to the web service and a location from which the document can be obtained by the web client (i.e. sharing the WSDL document to requesters outside the firewall, page 4 lines 1-4);

thereby enabling the web client to use the document to access the web service via the web service gateway (see page 4 lines 1-10, Fig. 1).

However, Venkatapathy does not teach where the profile containing the details of how to communicate is **in a format not recognizable to the web client**.

Chan, on the other hand, teaches a system in which the a profile containing the details of how to communicate can be formatted in other ways if it is not recognizable to the client by mapping CPA files (the profile definition in ebXML standards) into WSDL files (see pages 2,3, 8-10).

It would have been obvious to one of ordinary skilled in the art at the time of invention was made to modify the method in view of Venkatapathy teachings to include being in a format not recognizable to the web client taught by Chan. One ordinary skilled in the art wanting to publish a service using the gateway would look for ways to transform web service profiles from one definition to the other. One would be motivated to combine these teachings because there is a sufficient information in the CPA to generate WSDL definitions for all the parties involved.

Office Action dated April 30, 2008, pages 2-4 (emphasis in original).

The examiner bears the burden of establishing a *prima facie* case of obviousness based on the prior art when rejecting claims under 35 U.S.C. § 103. *In re Fritch*, 972 F.2d 1260, 23 U.S.P.Q.2d 1780 (Fed. Cir. 1992). For an invention to be *prima facie* obvious, the prior art must teach or suggest all claim limitations. *In re Royka*, 490 F.2d 981, 180 USPQ 580 (CCPA 1974). Independent claim 1 of the present invention reads as follows:

1. A method for a web services gateway to enable a web client to access a web service, the method comprising the steps of:

receiving, by a profiler in the web services gateway, a b2b profile from the web service, the b2b profile containing details of how to communicate with the web service and being in a format not recognisable to the web client;

creating, by the profiler in the web services gateway, a WSDL document based on the b2b profile, the WSDL document being in a format recognisable to the web client and containing details of how to communicate with the web service via the gateway; and

providing, to a third party, information relating to the web service and a location from which the WSDL document can be obtained by the web client;

thereby enabling the web client to use the WSDL document to access the web service via the web service gateway; and

wherein the details of how to communicate with the web service via the gateway include a RNIF/HTTP channel address in the gateway for the client to use when requesting to access the web service, the RNIF/HTTP gateway channel address being associated with the b2b profile and the method further comprises the steps of:

receiving a request, at the RNIF/HTTP gateway channel address, from the web client for the web service;

obtaining details from the b2b profile associated with the RNIF/HTTP gateway channel address and using the details to convert the request into a request suitable for sending to the web service; and

sending the converted request to the web service.

Neither *Venkatapathy* nor *Chan* teach or suggest the feature of a profiler in the web service gateway which receives a b2b (non-WSDL) profile from a web service and creates a WSDL document (a format recognizable to the web client) that contains details of how to communicate with the web service via the gateway. The examiner alleges that this feature is found in the following cited sections of *Venkatapathy* and *Chan* reproduced below:

Let us assume that you are providing a StockQuote service that is deployed inside the firewall of your enterprise and you want to share the service with your partners and customers.

#### **Step 1. Create the WSDL document**

Create the WSDL document that describes and invokes your StockQuote service. Listing 1 shows the sample of the WSDL that describes the StockQuote service. As you see, the soap address location is pointing to myhost that is inside the firewall.

*Venkatapathy*, page 2, lines 38-43.

Web services can be accessed from applications and processes both within the corporate firewall and beyond it. Fundamental to this is how the service, deep inside your enterprise network, is exposed to external users. This article looks at the issues involved in this approach and how IBM's Web Services Gateway addresses these issues.

*Venkatapathy*, page 1, lines 1-6.

### Agenda

- ◆ Scope of the project
- ◆ Current sub-team
- ◆ WSDL overview
- ◆ CPA -> WSDL element mapping
  - Rosettanet PIP3A4 example
- ◆ Open Issues

### Scope of the project

- ◆ Research how CPPA information can be integrated with following WS specifications\*
  - Web services description standard (WSDL)
  - Web service messaging standard/s (SOAP, SOAP + WS Routing + WS Security, etc.)
  - Web services choreography standard/s (IBM WSFL, MS XLANG, etc.)

\* in order of maturity /priority

*Chan*, pages 2-3.

### WSDL -> CPA element mapping

CPA	WSDL
<SimplePart> <NamespaceSupported location>	<import namespace, location>
<Packaging> <CompositeList>	<message>
<CanReceive action> <CanSend action>	<portType> -operations that the service can offer -operations that the service can invoke

### WSDL -> CPA element mapping

CPA	WSDL
<Packaging>	<binding> Assumption: Vanilla SOAP binding (use of attachments will require MIME binding)
<Transport Sender> <Endpoint> <TransportReceiver> <Endpoint>	<service> <port>

### CPA -> WSDL element mapping

- ◆ One WSDL document per CPA CollarationRole element
- ◆ No <wsdl:service> element for notification and solicit/response operations
- ◆ Preliminary conclusion: There is sufficient information in the CPA to generate WWSDL definitions for all the parties involved

*Chan*, pages 8-10.

Page 2, lines 38-43 of *Venkatapathy* discloses a scenario in a web services gateway for handling inbound web service requests. The cited section teaches an example StockQuote service that is deployed inside the firewall of an enterprise and the service needs to be shared with customers. The cited section also teaches that a WSDL document is created that describes and invokes the StockQuote web service. The SOAP address location for the service points to myhost that is inside of the firewall.

Page 1, lines 1-6 of *Venkatapathy* discloses that web services may be accessed from applications and process within the corporate firewall and beyond it.

Pages 2-3 of *Chan* disclose an agenda sheet listing agenda items including CPA (Collaboration Protocol Agreement) to WSDL element mapping, using Rosettanet PIP3A4 as an example, and defines a scope of a project of researching how CPPA (Collaboration Procotol Profile Agreement) information can be integrated with WS (Web Services) specifications, including WSDL.

Pages 8-10 of *Chan* disclose a table illustrating elements within a CPA and the mapped elements to WSDL, such that information in the CPA is used to generate WSDL definitions for the collaborating parties.

While *Venkatapathy* discloses a web services gateway and *Chan* discloses CPA to WSDL element mapping, there is no teaching or suggestion in either *Venkatapathy* or *Chan* of a profiler component in the web service gateway according to the presently claimed invention. The profiler, as shown in Figure 3 and described at least on page 9, lines 10-18, is a component in the web services gateway that receives a b2b (business-to-business) profile from a web service and creates a WSDL document that contains details of how to communicate with the web service via the gateway for those clients that do not recognize the b2b format. Neither *Venkatapathy* nor *Chan* discloses such a profiler component in the web services gateway. *Venkatapathy* merely teaches creating WSDL documents that describe the StockQuote service, and sharing the WSDL files to requestors outside of the corporate firewall. *Chan* merely discloses that CPA files may be mapped to WSDL files, but does not mention anything about a profiler component on the web services gateway, nor that such a profiler component receives a b2b profile from a web service and creates a WSDL document describing how to communicate with the web service via the gateway. Consequently, *Venkatapathy* and *Chan* do not teach or suggest a profiler in the web service gateway which receives a b2b (non-WSDL) profile from a web service and

creates a WSDL document (a format recognizable to the web client) that contains details of how to communicate with the web service via the gateway.

*Venkatapathy* and *Chan* also do not teach or suggest when a web client request is received at a RNIF/HTTP gateway channel address for a web server, details from the b2b profile associated with the RNIF/HTTP gateway channel address are obtained and used to convert the request into a request suitable for sending to the web service. This feature was generally recited in claim 3, and has been incorporated into claim 1. The examiner alleges that this feature is found in the following cited sections of *Venkatapathy* and *Chan* reproduced below:



```
<definitions targetNamespace="http://example.com/StockquoteService">
  <message name="GetQuoteInput">
    <part name="symbol" type="xsd:string"/>
  </message>

  <message name="GetQuoteOutput">
    <part name="quote" type="xsd:float"/>
  </message>

  <portType name="StockquotePT">
    <operation name="getQuote">
      <input message="tns:GetQuoteInput"/>
      <output message="tns:GetQuoteOutput"/>
    </operation>
  </portType>

  <binding name="SOAPBinding" type="tns:StockquotePT">
    <soap:binding style="rpc" transport="http://schemas.xmlsoap.org/soap/http"/>
    <operation name="getQuote">
      <soap:operation soapAction="http://example.com/GetTradePrice" />
      <input>
        <soap:body use="encoded" namespace="urn:mttoday:delayed:quotes" />
        <soap:encodingStyle href="http://schemas.xmlsoap.org/soap/encoding" />
      </input>
      <output>
        <soap:body use="encoded" namespace="urn:mttoday:delayed:quotes" />
        <soap:encodingStyle href="http://schemas.xmlsoap.org/soap/encoding" />
      </output>
    </operation>
  </binding>

  <service name="StockquoteService">
    <documentation>Stock quote service</documentation>
    <port name="SOAPPort" binding="tns:SOAPBinding">
      <soap:address location="http://myhost:8080/soap/service/rpcrouter" />
    </port>
  </service>
</definitions>
```

*Venkatapathy*, listing 1.

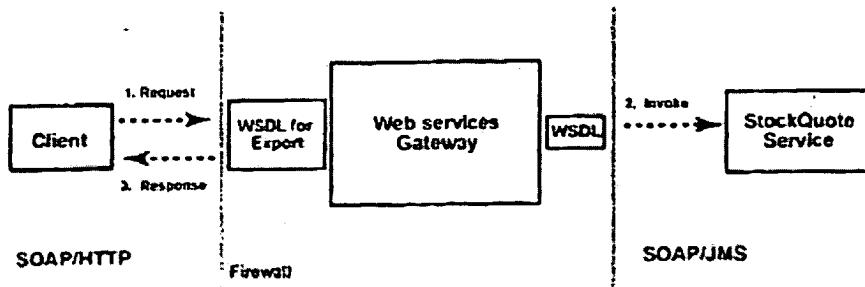
```

<definitions targetNamespace=">
<import namespace="http://www.ibm.com/namespace/wsif/samples/stockquote"
location="http://gatewayhost:80/wsgw/ServiceInterface?name=StockQuote"/>
<service name="StockQuote">
<port name="StockquotePTApacheSOAPBindingPort"
binding="interface:StockquotePTApacheSOAPBinding">
<soap:address location="http://gatewayhost:80/wsgwsoapi/soaprpcrouter"/>
</port>
</service>
</definitions>

```

Venkatapathy, listing 2.

Figure 3. Protocol transformation



Venkatapathy, Figure 3.

### Step 3. Share the WSDL document to requestors outside the firewall

This can be done in one of three ways:

- Requesting the Gateway to publish the service to UDDI, in which case service requestors may obtain it by using UDDI lookups.
- Using a copy of the WSDL obtained from the Gateway.
- Accessing a supplied URL that dynamically obtains the WSDL from the Gateway (See Listing 3).

Venkatapathy, page 4, lines 1-6.

### Scenario 3. Protocol transformation

Let us extend the Scenario 1. Your internal StockQuote service is available on SOAP/JMS and your customers and partners will invoke it on SOAP over HTTP. In this case, you mostly follow the same steps as scenario 1. However, in step 2, you specify in the Gateway that the service will be accessed by SOAP over HTTP. The Gateway will generate the new WSDL to share with your partners. In particular, the interface WSDL file will contain the SOAP/HTTP bindings rather than the original SOAP/JMS ones. Figure 3 shows the flow of inbound requests that gets transformed to SOAP/JMS invocations.

Venkatapathy, page 5, lines 1-7.

Listing 1 of *Venkatapathy* illustrates a sample WSDL document that describes the StockQuote service and the SOAP/HTTP transport channel for the service.

Listing 2 of *Venkatapathy* illustrates an interface WSDL file generated by the gateway specifying the gateway as the service end-point.

Page 4, lines 1-6 of *Venkatapathy* discloses sharing the WSDL documents to requestors outside the firewall by requesting the gateway publish the service to UDDI, using a copy of the WSDL obtained from the gateway, or accessing a supplied URL that dynamically obtains the WSDL from the gateway.

Figure 3 and page 5, lines 1-7 of *Venkatapathy* disclose a flow of inbound SOAP/HTTP requests that are transformed to SOAP/JMS invocations. The web service is available on SOAP/JMS and the web client invokes the service on SOAP over HTTP. As the service is invoked by SOAP over HTTP, the gateway generates a new WSDL to share with the web clients, and this interface WSDL file contains the SOAP/HTTP bindings rather than the original SOAP/JMS bindings.

*Venkatapathy* does not teach or suggest a b2b profile, nor of associating a b2b profile with a RNIF/HTTP gateway channel address. The presently claimed invention recites that when a web client sends a request for a service to the RNIF/HTTP gateway channel address, details from the b2b profile associated with the RNIF/HTTP gateway channel address are obtained and used to convert the request into a request suitable for sending to the web service. While *Chan* discloses CPA to WSDL element mapping and using Rosettanet PIP3A4, *Chan* does not mention anything about receiving a request at an RNIF/HTTP gateway channel address and obtaining a b2b profile of a service associated with the RNIF/HTTP gateway channel address. *Venkatapathy* merely discloses receiving SOAP/HTTP requests at a gateway from a client and transforming the SOAP/HTTP requests to SOAP/JMS invocations using original and new interface WSDL documents. There is no mention in *Venkatapathy* of associating a b2b profile with a particular location or gateway channel address in the web services gateway, or of using this association to obtain details from the b2b profile of a web service. Consequently, *Venkatapathy* and *Chan* do not teach or suggest that when a web client request is received at a RNIF/HTTP gateway channel address for a web server, details from the b2b profile associated with the RNIF/HTTP gateway channel address are obtained and used to convert the request into a request suitable for sending to the web service.

Claims 2 and 4 are dependent claims depending on independent claim 1. Dependent claims 2 and 4 are also not obvious over *Venkatapathy* in view of *Chan*, at least by virtue of their dependency from claim 1.

Therefore, the rejection of claims 1-4 under 35 U.S.C. § 103 has been overcome.

**II. Conclusion**

It is respectfully urged that the subject application is patentable over the cited references and is now in condition for allowance.

The examiner is invited to call the undersigned at the below-listed telephone number if in the opinion of the examiner such a telephone conference would expedite or aid the prosecution and examination of this application.

DATE: September 29, 2008

Respectfully submitted,

/Cathrine K. Kinslow/

Cathrine K. Kinslow  
Reg. No. 51,886  
Yee & Associates, P.C.  
P.O. Box 802333  
Dallas, TX 75380  
(972) 385-8777  
Attorney for Applicants